

# APPARATUS AND METHOD TO AUTOMATICALLY ACTIVATE A RESERVE RESOURCE

## Field of the Invention

5 The invention pertains to an apparatus and method to monitor a load on a number of active resources and to automatically activate a reserve resource when the load on the number of active resources meets a threshold specified in a resource usage policy.

## Background of the Invention

10 Computing and network resources are subject to wide fluctuations in demand. For example, a computing system or a component thereof (e.g., a CPU or memory) can experience fluctuations from little to no load in the early morning hours or on weekends, to very high loads during late afternoon, weekday usage, or during the evening hours when many computing systems  
15 are programmed to perform high volume calculations and other data processing. Likewise, user demand on Internet or other network resources (e.g., a server or host computer) depend on a variety of factors such as the number of users or the time of day. In addition, traffic to a particular Internet site  
20 can start out slow and due to advertising, linking, etc., grow exponentially in a short period of time. The timing of such increases in demand are often difficult

to predict, making it difficult to anticipate or plan for the computing and/or network resource capacity which is necessary to meet the demand at any given time.

One solution is to provide more resource capacity (e.g., more CPUs, more servers, etc.) than is estimated to be necessary for normal conditions. However, such over-allocation of resources can be costly and unnecessary. In addition, many small or start-up companies may not be able to afford extra resources until, for example, sales volume increases. Therefore, another solution is to provide both active resources and reserve resources, where the active resources are sufficient for typical demand and the reserve resources are only activated when there is a peak in demand (i.e., "capacity-on-demand"). However, current approaches to capacity-on-demand require manual intervention to activate an additional resource. That is, before the active resources reach capacity, the user must contact the vendor and obtain a code to "unlock" and activate the additional resource. Therefore, an administrator must be "on-call" at all times during operation (i.e., twenty-four hours every day, seven days a week) to monitor the active resources and activate reserve resources as necessary. In addition, sudden spikes in demand that require immediate deployment of an additional resource can cause interruptions in service before the additional resource can be activated.

It is therefore a primary object of this invention to provide the resource capacity that is necessary to meet fluctuations in user demand dynamically or automatically. Another object of the invention is to automatically activate a reserve resource when additional resource capacity is required to meet demand.

### **Summary of the Invention**

In the achievement of the foregoing objects, the inventors have devised an apparatus and method to monitor the load on a number of active resources

and to automatically activate a reserve resource when the load on the active resources meets a threshold that is specified in a resource usage policy.

The invention can be embodied in software or firmware (i.e., program code), stored in a computer readable storage media (e.g., a disk, a RAM, or other memory). The software can be executed, for example, by a computer processor linked directly, over a network, through a peripheral device, etc. to an active resource (or pool of active resources) and a reserve resource (or pool of reserve resources). A resource usage policy is stored in the storage media or memory and includes at least one threshold for triggering at least one task. For example, the threshold of 90% and the task of activating a resource can be specified in a resource usage policy as "If Active Resource No. 1 usage exceeds 90% for more than ten seconds Then Activate Reserve Resource No. 1." Tasks might also comprise transmitting a pager alert, generating an alarm, and so on. A computer readable program code is also stored in the memory and is accessed by the processor for performing a number of functions. These functions include monitoring the load on a number of active resources such as a CPU or a server and comparing the monitored load (e.g., as a function of percent utilization and/or duration of usage) to the threshold specified in the resource usage policy. When the resource usage policy dictates, corresponding task is performed.

As an example, when the load meets the threshold specified in the resource usage policy, the processor can send an alarm to the administrator's pager to warn the administrator that the active resource has reached the threshold, or the processor can automatically activate a reserve resource. Preferably, a hierarchy of monitoring policies (i.e., specified thresholds and corresponding tasks) are used for different levels of resource allocation. For example, at a specified threshold of "90% usage for ten seconds," the corresponding task is a "pager alert," and at a specified threshold of "95% usage for five seconds," the corresponding task is to "activate an additional CPU," and at a specified threshold of "95% usage of one or more system components for ten seconds," the corresponding task is to "activate an entire

09709705 110900

system." Alternatively, the administrator and/or the processor can dynamically set thresholds and/or tasks based on any suitable factors such as the rate of resource consumption, the number of remaining reserve resources, etc. For example, when the monitored rate of resource consumption is "an additional 1% active resource consumption per hour," the processor and/or administrator can set the next task to "generate a pager alert," whereas when the rate of resource consumption is "an additional 1% active resource consumption per minute," the processor can set the next task to "automatic activation of a reserve resource."

Once a reserve resource is activated, the program code generates an indication that the reserve resource is activated. The indication can trigger a pager or email alert sent to an administrator, a signal sent to a load balancer or other peripheral device, updating a configuration profile, etc. Preferably, the program code also causes the processor or a peripheral device such as a load balancer to balance the load among the active resource and the activated reserve resource. In one embodiment, the processor or peripheral device also transfers at least part of the load from the number of active resources to the activated reserve resource, thereby reducing the load on the active resources. Preferably, the processor or a peripheral device also maintains a configuration profile for each active resource, each activated resource, and each reserve resource. The configuration profile can be used to optimally allocate the available resources, and to alert a system administrator before the system reaches an over-committed state (i.e., where there are insufficient reserve resources to handle the demand). In addition, once the load meets a specified threshold (i.e., the activation threshold or an altogether separate threshold), the processor preferably deactivates the reserve resource, thereby reducing operation costs and returning the reserve resource to the reserve resource pool. The resource is then once again available for future allocation. Another embodiment of the invention includes charging a fee for monitoring and activation.

As such, the apparatus and method of the present invention provide resource capacity-on-demand to meet fluctuations in user demand without

unnecessary and costly over-allocation of resources. In addition, the reserve resource is automatically activated so that manual intervention is not required, thereby reducing or eliminating service interruptions and the need for an administrator to be on-call during operation. However, the present invention also preferably alerts an administrator to give him or her the opportunity for manual intervention and/or monitoring when desired. The load continues to be monitored after activating the reserve resource so that the activated reserve resource can be deactivated when it is no longer needed, or so additional reserve resources can be activated if necessary. Thus, the reserve resource is automatically returned to the reserve resource pool, thereby reducing costs (i.e., the user need only pay for the time that the resource is allocated) and/or making the reserve resource available to other systems for efficient and optimum resource allocation. In addition, the invention also includes a new utility model for doing business in which the purchaser pays for service up front, but does not pay for the hardware until it is actually needed.

These and other important advantages and objectives of the present invention will be further explained in, or will become apparent from, the accompanying description, drawings and claims.

#### **Brief Description of the Drawings**

An illustrative and presently preferred embodiment of the invention is illustrated in the drawings in which:

FIG. 1 is a block diagram of the components of the monitoring and activation apparatus;

FIG. 2 illustrates a pool of resources for use with the monitoring and activation apparatus;

FIG. 3 illustrates monitoring policies for use with the apparatus and method of the present invention;

FIG. 4 is a flow chart showing steps to monitor a load on an active

resource and perform a corresponding task when the load meets a threshold;  
and

FIG. 5 is an exemplary configuration profile.

5

### **Description of the Preferred Embodiment**

FIG. 1 shows various hardware components which can be used by the monitoring and activation apparatus 10 of the present invention. The apparatus 10 includes a processor or CPU (central processing unit) 20 and a computer readable storage device or memory 30. Optionally, the processor 20 can be linked over a conventional communication or other network link directly to a pager, phone, computer or the like, or indirectly to the above via an event manager 50. The processor 20 is also linked through an optional peripheral device 40 such as a load balancer to an active resource 60 (e.g., a CPU or a server) or an active resource pool 70 having a plurality of active resources 60, 61, 62 therein. The processor 20 is also linked through the same or another optional peripheral device 40 to a reserve resource 80 or a reserve resource pool 90 having a plurality of reserve resources 80, 81, 82 therein. Computer readable program code is provided (e.g., stored in memory 30 and accessed by the processor 20) for monitoring the load on the active resource 60 (or pool 70) and comparing the monitored load to a threshold 315 (FIG. 3) specified in a resource usage policy 310 (FIG. 3) that can be stored in memory 30. When the monitored load meets the specified threshold 315, the program code dictates performance (e.g., by processor 20, peripheral device 40, etc.) of a task 320 (FIG. 3) corresponding to the specified threshold 315. For example, the corresponding task 320 can be to signal the event manager 50 or to activate a reserve resource 80 from the reserve resource pool 90. Once a reserve resource 80 is activated, the program code preferably dictates generating an indication that the reserve resource 80 is activated. The indication can be an alert sent via an Internet HTML (hypertext mark-up language) page, a pager,

email, etc. to the administrator. The indication can also be a signal sent to a peripheral device 40, or a combination alert and signal. Preferably, the program code also dictates that the processor 20 (or a peripheral device 40) balance the load among the active resource 60 (or pool 70) and the activated reserve resource 80. When the load drops below the specified threshold 315 (or meets a separately specified threshold 315), the program code also preferably dictates that the reserve resource 80 be deactivated and returned to the reserve resource pool 90. Also in a preferred embodiment, a configuration profile (e.g., a database in memory 30 containing each resource and the status thereof) is updated (or even maintained) for tracking the status of the active resource 60 and the reserve resource 80 (and optionally pools 70, 90).

It is understood that the processor 20 can be any conventional processor such as an Intel PENTIUM® processor, a personal computer system (PC) such as a desktop or laptop computer, or can be an ASIC (application specific integrated circuit) which is specifically manufactured for use with the present invention. Likewise, the computer readable storage medium or memory 30 can be any suitable memory such as RAM, ROM, magnetic tape, compact disc, a combination thereof, etc. The processor 20 and the memory 30 need not be separate units and can be combined, or alternatively, the processor 20 and the memory 30 can be separately housed and linked to one another over a remote network or other suitable connection. Likewise, there can be a number of (i.e., one or more) processors 20 and/or a number of (i.e., one or more) memories 30 that are connected or linked via the Internet, Intranet, LAN, WAN, etc. In such a scenario, the storage of the aforementioned program code may be distributed over said memories, or executed by various of the processors.

The event manager 50 monitors system resources, configures monitoring requests, checks resource status, and sends notification when configured conditions are met. An example of an event manager 50 is described in the "Event Monitoring Service User's Guide," HP Part Number B7612-90015, November 1999, Copyright 1999 Hewlett-Packard Company). It is also understood that the event manager 50 can pass an event notification to any

suitable device to notify the administrator that the specified threshold 315 (FIG. 3) has been met. In addition, the event manager 50 can pass an event notification to notify a vendor that the specified threshold 315 (e.g., a CPU has been activated) has been met so that the vendor can monitor customer compliance with billing policies, charge the user a fee when a reserve resource 80 is activated, etc. For instance, the event manager 50 can pass an event notification to a conventional pager, a cellular or other handheld phone, an Internet appliance, a desktop or laptop computer, a conventional display, signal light, audio speaker, an Internet HTML page, an e-mail address, a combination thereof, etc. Similarly, the event manager 50 can be hardwired to the processor 20 or can be accessed by the processor 20 via a remote connection (e.g., via the Internet, a telecommunications network, a modem connection, a T-1 or DSL connection, satellite, etc.). It is to be expressly understood that these examples are merely illustrative and are not intended to limit the scope of the invention.

FIG. 2 is illustrative of various types of resources 60, 80 which may be active or on reserve in the system of FIG. 3. An active resource 60 or reserve resource 80 can be an individual component 200-208 and 211-219 such as a CPU, a memory, a peripheral device, etc., and/or can be an entire system 210, 225 such as a host computer or PC, a server, etc. Likewise, the active resource 60 and the reserve resource 80 can be a combination thereof, such as a CPU, a memory, and a PC system. A number of (i.e., one or more) resources 60, 80 can be monitored as dictated by the program code and executed on a processor 20 directly connected to the resources 60, 80, or connected to the resources 60, 80 over a conventional network 220.

It is understood that the arrangements shown in FIG. 1 and FIG. 2 are merely illustrative and not intended to limit the teachings of the present invention. For example, the resources 60, 80 and processor 20 (and related memory 30, peripheral devices 40, and event managers 50) need not be clustered. That is, the resource pools 70, 90 can include resources 60, 80 that are dispersed, for example, over the network 220. Likewise, the processor 20 and/or peripheral device 40 can control (e.g., monitor, activate, deactivate) the



resources 60, 80 in more than one resource pool 70, 90.

FIG. 3 illustrates monitoring policy (e.g., stored in memory 30). The monitoring policy 300 includes at least one resource usage policy 310 specifying a threshold 315 therein and having a task 320 corresponding thereto. However, it is understood that the monitoring policy can include any number of resource usage policies 310-312 (Policy 1, Policy 2,... Policy n), each specifying one or more thresholds 315-317 therein and can have any number of corresponding tasks 320-323 (Task 1A, Task 1B, Task 2,... Task n) assigned or corresponding thereto. In addition, more than one resource usage policy 310 can trigger the same task 320 (not shown), and more than one task 320 can be triggered by a single resource usage policy 310 (e.g., Task 1A and 1B both correspond to Threshold 1 in FIG. 3).

Preferably, the monitoring policy includes a hierarchy of resource usage policies 310 with thresholds 315 and tasks 320. That is, a low threshold can be set to recognize the early signs of high demand on the active resource 60, and the processor 20 can alert the administrator of these early signs of high demand. Thus, the administrator can manually intervene at an early stage (i.e., before additional resources are activated and a fee charged for their use). For example, the administrator can check whether the high demand is due to increased user traffic or a failure in a software routine (e.g., an endless loop) that has caused a resource to be over-utilized. In the example where there is a failure, the administrator can manually intervene to reset the application causing over-utilization. Or in another example where the administrator decides to allocate low cost resources instead of the higher cost resources designated by the task 320 corresponding to the next specified threshold, the administrator can manually allocate these alternative resources. A higher threshold can also be set, so that where the resource utilization continues to increase after the alert has been sent to the administrator, the program code can dictate activating the reserve resource 80, and, if desired, also transmit a message to the administrator notifying the administrator that the reserve resource 80 has been activated. As such, the administrator can check if the reserve resource 80 was

activated in error or as part of a test and deactivate the reserve resource if necessary (e.g., before incurring a charge). Another still higher threshold can be specified to activate an entire system as the load continues to increase. Yet a higher threshold can be specified accounting for the combination of active and activated reserve resources 60,80 when there are no additional resources (i.e., the system is entering an over-committed state). When this yet higher threshold is met, the program code can dictate transmitting a high priority alert to the administrator notifying the administrator that the load is continuing to increase and that absent manual intervention such as closing applications, the resources may fail. An example of such a hierarchy is shown in the following table.

TABLE I

Threshold	Task
1) Average CPU usage > 90% for 10 seconds	1) Transmit Alarm "High CPU Usage"
2) Average CPU usage > 95% for 5 seconds	2) Activate Reserve Resource AND Transmit Alarm "High CPU Usage - Reserves Activated" AND Balance Load
3) Average Combined CPU usage > 80% for 10 seconds	3) Activate Reserve Resource (Entire System)
4) Average Combined CPU usage > 80% for 10 seconds AND Additional Reserve Resources Unavailable	4) Transmit Alarm "High CPU Usage - No More Reserves To Activate - Manual Intervention Required"

It is to be expressly understood that the threshold 315 can be 1) set (i.e., specified in the resource usage policy 310) prior to monitoring, 2) set or changed during monitoring, or 3) set based on a combination thereof. That is, the threshold can be specified in the resource usage policy 310 prior to the load reaching the threshold. However, the program code can dictate, and/or the

administrator can manually specify, a threshold 315 during monitoring. That is, a threshold 315 can be specified and/or assigned a corresponding task based on various factors such as the rate of resource consumption, the number of remaining reserve resources, or any other suitable algorithm.

5           As an example, when the rate of resource consumption is "an additional 1% active resource consumption per hour," the task "generate a pager alert" can be assigned to the next specified threshold 315, whereas when the rate of resource consumption is "an additional 1% active resource consumption per thirty seconds," the task "automatic activation of a reserve resource" can be  
10 assigned to the next specified threshold 315. In another example, the program code can dictate that a status be transmitted to the administrator upon reaching a first specified threshold 315, and the administrator can then specify a second threshold 315 and corresponding task 320 based on the status and/or other considerations. In yet another example, the administrator can transmit one or  
15 more resource usage policies 310 before or during monitoring via keyboard input, e-mail, FTP (file transfer protocol), voice commands, etc. Indeed, entire monitoring policy can be transmitted by the administrator at any time prior to or during the monitoring. Likewise, resource usage policies 310, including thresholds 315 and tasks 320, can be established in the monitoring policy 300  
20 as defaults.

FIG. 4 shows a method to monitor the load on an active resource 60 and perform a corresponding task 320 when the load meets a threshold 315 specified in the resource usage policy 310. In step 400, the program code dictates monitoring a load on an active resource 60. For example, where the  
25 active resource 60 is a CPU or server, the load can be percent usage over time. Or for example, where the active resource 60 is a memory, the load can be space utilization with respect to space remaining. Any suitable algorithm for monitoring the load can be used under the teachings of the present invention and can be done using conventional component or system auditing software  
30 (e.g., the Microsoft WINDOWS® Resource Meter). The program code also dictates comparing the monitored load in step 410 to a threshold 315 specified

5  
10

15

20

25  
30

resource as an available or activated reserve resource. Activation API (Application Programmers Interface) software is currently available to functionally activate hardware when a user inputs a predetermined passcode. That is, when the administrator recognizes that additional capacity must be added to the system, the administrator contacts the vendor and receives a passcode. The administrator then inputs the passcode to the activation API, which then "unlocks" the resource. Therefore, in one embodiment of the present invention for automatically activating reserve resources, upon detecting that a reserve resource 80 must be brought online (i.e., during monitoring in step 400), the program code can either "look-up" a passcode stored in memory 30 (e.g., in a look-up table or other data structure), or the program code can cause a connection to be established with the vendor (e.g., modem dial-up connection, DSL, etc.), and download or otherwise retrieve the passcode. The program code then submits the passcode to the activation API to activate the reserve resource 80. The appropriate threshold 315 can be specified to account for delays in obtaining and submitting this passcode. However, in another embodiment, a passcode is not necessary, and the activation API can be programmed to respond to other activation requests or signals made by the program code. It is understood that the activation API can be integrated into the OS, an overlay to the OS, etc. Alternatively, the reserve resource 80 can be electrically inactive at startup, and can be made electrically active prior to use. Program code or firmware to activate an electrically inactive resource is also within the understanding of one skilled in the art (e.g., a computer BIOS or OS power management functions).

In step 442, the program code dictates indicating that the reserve resource 80 has been activated. The indication can be a message sent from the processor 20 to the administrator via an event manager 50, a signal sent to the peripheral device 40, a signal sent to a subroutine within the program code, a combination thereof, etc. In step 447, the program code dictates balancing the load among the active resource 60 and the activated reserve resource 80. For example, the load balancer 40 can route new tasks to the activated reserve resource 80. Alternatively, the load balancer 40 can transfer at least part of the

load from the active resource 60 to the activated reserve resource 80 and then route new tasks to the resource 60 or 80 that is better able to handle the new task based on processing requirements, time requirements, etc. In step 460, the program code dictates updating a configuration profile 500 (FIG. 5). The configuration profile 500 can be a database stored in memory 30, such as that shown in FIG. 5, containing the status 510 (e.g., active, inactive, load, capacity, etc.) of each resource 60, 80 specified therein (e.g., at 520). The configuration profile 500 can be used to at least track the active resource 60 and the reserve resource 80 (both activated and inactive). As such, the load balancer 40 is able to route or balance the load among the available resources 60, 80, to activate additional reserve resources 80 when necessary, to generate alarms when there are no more reserve resources 80 available, etc.

As yet another example, the task 320 can be defined as "deactivating a reserve CPU" when the monitored load on the active resource 60 and/or the activated resource 80 is "less than 85% CPU usage for fifteen seconds" (i.e., the specified threshold 315). Thus, when the monitored load meets the specified threshold 315, the program code optionally dictates returning any remaining load on the activated reserve resource 80 to the active resource 60 in step 450 (FIG. 4) and deactivating the reserve resource 80 in step 455, again using control software and hardware as explained above with respect to activating the reserve resource. As such, the reserve resource 80 is returned to the reserve resource pool 90 and can be powered down to save cost, be made available to other systems, etc.

It is to be understood that the specified threshold 315 corresponding to the task 320 of deactivation can be the same specified threshold 315 as that corresponding to the task 320 of activation. However, preferably, the specified threshold 315 corresponding to the task 320 of deactivation is a separate specified threshold 315 to reduce bounce (i.e., switching a reserve resource 80 on and off) when the load on the active resource 60 fluctuates near the specified threshold 315. For example, the specified threshold 310 corresponding to the task 320 of deactivation is set so that the load must drop

It is to be understood that any suitable task 320, threshold 315, or resource usage policy 310 can be dictated by the program code at 424 in the routine illustrated in FIG. 4. For example, the program code can dictate swapping a reserve resource 80 with an active resource 60 when the active resource 60 begins to fail (e.g., the threshold “processing error returned” is met) or when the active resource 60 should be upgraded (e.g., the threshold “processing speed below X MHz” is met). In addition, in a preferred embodiment, the program code dictates continued monitoring of the load after performing a task 320 so that any further changes in the load can be detected (i.e., the specified threshold 315 is met) and acted upon (i.e., the corresponding task 320 is performed).

20

25

30

tasks 320 can be included at 424, or step 450 can be omitted when there is no load on the reserve resource 80 to be deactivated in step 455.

Another embodiment of the invention includes automatic monitoring and activation pricing. In one embodiment, a basic fee is charged for a system (e.g., a PC, a server, etc.). The basic fee includes the fee for the resources that are anticipated to be used on a regular basis (i.e., activated resources 60). The user is then charged a fee for use of reserve resources 80 using at least one of three different charge options. One option includes charging the basic fee in addition to a "one-time" use fee that is charged once a reserve resource 80 is activated. Under this option, once the reserve resource 80 is activated, the reserve resource 80 is purchased and no additional fees are charged for later use. A second option includes charging a basic fee in addition to an "as-needed" fee. Under this option, the "as-needed" fee is only charged for the period of use of the activated reserve resource 80. In this case, the basic fee for a reserve resource is essentially an option fee. Once the reserve resource 80 is deactivated and returned to the user pool 90, there are no additional charges. A third option includes charging a fee only upon activation of the reserve resource 80. Under this option, no basic fees are charged and the charges are only made on an "as-needed" basis as described above with respect to the second option. These charge options can be included in the monitoring policy and dictated by the program code (e.g., as tasks 320). For example, a policy might state "If load >90%, activate resource and alert billing agent that resource is being used." The above three illustrations are merely illustrative and other embodiments are also contemplated under the present invention, such as but not limited to charging a fee for administrator alerts, etc.

While illustrative and presently preferred embodiments of the invention have been described in detail herein, it is to be understood that the inventive concepts may be otherwise variously embodied and employed, and that the appended claims are intended to be construed to include such variations, except as limited by the prior art.

09709705 110500